

December 2, 2007

To: CPB  
From: Amber  
Re: "Charge to SAMGrid-DØ Project Manager"

**Observations:**

Given that we are now six months into the DØ grid initiative, I offer the following observations.

- 1) We are making steady, but slow progress on transitioning the SAMGrid operations from the development team to an operations team and that transition has been both more work and taken longer and has been more disruptive than the original estimates.
- 2) We are making steady progress on using SAMGrid for primary farm production, but for the reasons listed above, this has not been a smooth process. Some of this is normal, inherent in rolling out a new system, some of it has illuminated a lack of good process for having understood reproducible configurations and understanding of a complicated system. We have been working to address this.
- 3) People are extremely good at reacting to problems, which is admirable in and of itself, but the desired state for a system in operations is that problems rarely occur. We are not at that state for either MC production or for primary production. There is no single element that is causing this—it's a fact of life given the current state of grid middleware.
- 4) The timescale for Run II is more uncertain which does not make planning easier.
- 5) Many of the minor tasks on the CPB list have been completed.
- 6) Many of the items on the work list involved changes to the merging, either to move it to a more natural location in the workflow or to handle more complicated cases. At the current time, no experiment appears to have generically solved the merging problem—we should not underestimate the amount of work required to re-implement merging.
- 7) We have to come to grips with the fact that in any system we have for running production, people will have to be assigned/found to run the production. We are not going to achieve some automated nirvana.
- 8) It has been significant operational stress to convert the primary production to use SAMGrid while the current script systems for skimming, CAF etc. are stable, flexible and require low labor for operations.
- 9) It has been extremely difficult to schedule work
  - a. If multiple people or teams are involved, it's been practically impossible
  - b. There are very few people and so a lot of work proceeds serially and any slippage in one area unavoidably leads to delay in others, particularly when it is necessary to react to operational problems.
  - c. It has been difficult to find people to do the testing on the DØ side

- d. If we were going to use the DØ grid system for executables beyond what we current use, there is work which has to be done on the DØ side to prepare the runtime environment executables/rcps/inputs for a generic style of production running.

Now, to the items in the charge:

1. *To enumerate the projects required in order to get DØ datahandling to a state of operational stability for the balance of Run IIb.*

DØ computing is operationally stable however based on the task list produced by the CPB, (see [SAMGrid Priority Discussion — DØWiki](#)) there are a number of areas of possible improvement. The projects required to improve stability, robustness or reduce operational load can be divided into several classifications. In this document, I have attempted to group the requests into local groups and in general, the items listed below reflect items on the CPB list or related work. Looking at the requests case by case is provided in another section.

1. Projects which must be undertaken due to excessive risk or effort or to facilitate other projects. There are three projects which fall into this classification

- a. Transitioning SAMGrid operations to an operations team. This transition is a project in itself due to the development needs of reducing operations.
- b. Transitioning primary production to use the OSG interfaces on local FNAL resources. This addresses the obsolescence of dFarm and fbsng products and increases flexibility for using the hardware resources and provides possible mechanism for an extension to other applications. Further it supplies us with a “live” example of using OSG resources for data reconstruction which will facilitate a fixing pass using non-local resources.
- c. Redefining the interfaces/responsibilities between d0runjob and SAMGrid such that integration of executables into the workflow could potentially take place independently of the integration with SAMGrid. There are several components of this work—the first phases address the scripting control of the workflow with a possible redistribution of file handling responsibilities (such as merging) in the future.

2. Projects which address the many important parts of the data chain that are in the hands of an individual collaborator for maintaining the scripts, and running the production and as examples, most of these executables are on the CPB list.

- a. Primary reconstruction (addressed above)
- b. skimming
- c. CAF tree production
- d. MC CAF tree production
- e. recocert
- f. AADST
- g. raw2sim

3. Feature augmentation to address limitations

- a. Ability start or store MC at any stage.
- b. Minimal resource brokering
- c. Issues associated with file delivery, merging or storage.

4. Projects which are completely at the discretion of the DØ collaboration once the workflow changes are implemented and tested. An example of such a project is to get TrigSim into the Monte Carlo chain.

#### **Timescales:**

For timescales, the assumption is that the collider data chain of reconstruction, skimming, and CAF tree production will end roughly on the same timescale of data collection. That would mean that the tools and applications have a further lifetime of 3-4 years. It is possible that a post run fixing pass will be required which suggests that it is prudent to assume a somewhat longer lifetime of these same tools.

Monte Carlo production will continue for long after the end of data collection, giving it a lifetime of 4-5 years.

It is almost certain that a “fixing” pass will be required in 2008; and it is possible that grid resources will be required.

Most of the possible projects described in this note would be executed by a very few individuals, all of whom are currently overcommitted with operations, physics analysis or other demanding tasks. For this reason, the record over the past year for being able to predict the amount of time it takes to do any given project has been underestimated, frequently dramatically.

#### *To analyze the SAMGrid-DØ product architecture*

The analysis was performed by Adam Lyon and can be found in several documents. In this instance, “architecture” refers to roles, responsibilities and boundaries between the set of services required to configure and run a DØ application (or set of applications) known as DØ runjob, the file data handling system, services required for job submission through the grid interfaces and execution and monitoring, and a high level layer for (semi)-automatic submission and recovery of bulk requests (d0reprotools or automc). In general, the discussion centers on the interfaces between d0reprotools, JIM (the grid job submission facility) and d0runjob. Ideally, the JIM layer would be self contained from the experiment applications. In practice, there are several couplings caused by need for efficiency and expediency.

#### *2. To analyze the cost-benefit-risk tradeoff for each project identified in 1 and 2.*

**The first class of projects** listed (transition of SAMGrid to operations, workflow autonomy for RunJob; primary production using SAMGrid) are essential—the benefits include enabling the possibility of pursuing the other projects. In general—it is not possible to contemplate doing any major task on the CPB list without doing those three

projects. Thus cost and risk are not relevant concepts. Completion of these projects defines the start of others. In addition, the SAMGrid development team also has responsibility for Monte Carlo production, which limits the rate of progress.

**The second class of projects** (addressing the owner/operator scripts and other non-production aspects in making data sets) for the applications needed to produce datasets is considerably more complicated.

There are three primary risks to continuing as we have for the past 5 years

- a. Individuals may no longer being able or willing to maintain their scripts/procedures.
- b. The systems depend on the availability of local resources tailored to DØ applications.
- c. Finding the effort to continue to run six different applications by six different single individuals. On the other hand, it might be easier to find six people to spend an hour a day on one focused task than to find four people who are willing to cross train and spend six hours a day on their shift to run multiple productions.

The cost of continuing as in the past is slight—most of the applications take about an hour a day or less to run, and it is unlikely that operational load can be reduced by much. The cost of maintaining the system is similar—someone will have to do it.

Additionally, at this time CAF trees are remade much more often than the skims and the CAF trees are not always remade at the same time for all skims. It may be that private “production” remains the most satisfactory solution.

There are three general possibilities for changing the situation. Not all possibilities address all risks. Essential to all possibilities is to insure the application runs out of a release (no private executables, no private rcps and no private runtime files) such that if necessary, we can move away from the need for DØ tailored resources.

- a. “Productize” the existing setups and documenting their usage to enable “farmers” to run them would address the third risk and mitigate the first one. It also has the advantage of being relatively low cost. If the standard DØtools/caf\_tools is used, it may be possible to mitigate the risk of relying on DØ specific resources. It must be stated that much of the work that has to be done in order to use the d0repro-tools/SAMGrid/runjob system is on the DØ side to prepare the executables to run in a generic environment. Thus, “productizing” is a first necessary step.
- b. Introduce applications individually into the current production system of d0repro-tools-SAMGrid-d0runjob in an ad hoc way, such as complete the prototype support for skimming in SAMGrid and d0repro tools. In this solution we would have to choose very few of the applications, and would likely would increase the risk and fragility of the production chain. An example from the past is the solution of shoehorning production of reco-cert output into the merging step—it was expedient, but complicated operations. There is very high barrier for solutions which cause increased operations or maintenance.
- c. Continue working on the recommendations in Adam Lyon architecture documents, with a priority which best enables the collaborators to use the

production system. There is a risk for this course of action that the amount of work to make this truly feasible exceeds the available resources and the time. Further, there is no certainty that the DØ experimenters who currently maintain individual systems will want to invest the effort to use the production system.

Solution b is highly disfavored—we already have ad hoc scripted solutions for all of the applications; why introduce new ones at this stage of the experiment? This means that the cost of moving any of the applications to a different solution involves elements of a comprehensive solution and it implies application preparation such that the applications can be run on non DØ tailored resources. One implication of this statement is that chains requiring merging are tied up with moving from a limited implementation of merging in the current SAMGrid to a more general solution implementation.

*Make Recommendations:*

**The first recommendation** is to complete the essential tasks as listed above. In general the responsibility for those tasks rests with the project managers for the individual products (such as the SAMGrid project leader), with the tasks tracked as part of an umbrella project. There is a WBS structure for this work and much progress has been made.

**The second recommendation** is to productize the existing scripts used for skimming and caf tree production, raw2sim and aadst. If the current mechanism of running recocert as part of the merging step doesn't cause any operational problems, then we can continue doing that. Productizing would give us the best ability to reduce risk while keeping avenues open for eventually using the production system.

**The third recommendation** is, if it is consistent with timescales and efforts, to determine the changes to SAMGrid and d0reprotools required that would best support skimming. Ideally, this work would have been done as part of the background for this document, in practice the people who need to do the technical evaluation have been occupied with the transition to operations.

Addressing the individual items:

## Monte Carlo

### 1. Ability to write out any stage of output.

Status: runjob implementation should make this possible.

Attached Documentation: none

MC production involves pushing events through a series of programs such as GEANT simulation (d0gstar), digitization/zerobias overlay (d0sim), and reconstruction (d0reco). Currently we only save into SAM the final output stage (thumbnails). The other stages are only declared. We want to add the ability to merge and store to tape some/all of these intermediate stages on demand. An example is to store the output of the GEANT stage since it is generally the most CPU intensive stage. Initial tests revealed the need for some further SAMGrid development. Current estimate

(from Parag) is one day of code writing and 3-4 days of testing. It is believed that writing out d0gstar output now , since merging is unnecessary. Testing of this is to come.

**Assessment of the CPB analysis:**

This was clearly necessary. The functionality has been provided and as of Nov. 1 requires testing.

**Costs:**

**Benefits:**

**Risks:**

**Comments:**

---

**2. Ability to start production using a SAM definition as input files for any stage of production.**

Status: not started

Attached Documentation: none

There are two primary examples of why this is useful. First, some event generators are not well suited for farm production. CompHEP and Sherpa are such generators. We can generate the events on our own computers, store them into SAM and then have the farms process the events through the later stages. Second, use events from an intermediate stage from another request for further processing. For some processing, we might save the GEANT output and later start a new request from it while doing different zerobias overlay. The first case originally worked with SAM v5 and was critical for our single top result, but it currently does not work. Time estimate needed from Parag.

**Assessment of the CPB analysis:**

The functionality from v5 has been replicated after negotiation with the MC team.

**Costs:**

**Benefits:**

The benefits of the more general case are unclear as no one was able to make a definite case beyond the "might".

**Risks:**

**Recommendation:**

Revisit general case after the more pressing projects are completed.

---

**3. Support testing of functionality of v7/d0runjob/SAMGrid as needed.**

Status: ongoing

Attached Documentation: none.

This is the route to getting rid of SAM v5. Over the next few weeks, DØ people led by the Monte Carlo Request Coordination team, will test that the full functionality of these three key components work. Examples of such functionality would be the ability to use different generators (such as Alpgen) or additional programs such as d0mess or EvtGen. At this time, Alpgen is working. If problems are found in other tests, the d0runjob developers and SAMGrid developers will work together to identify solutions. It is still difficult to estimate this time.

**Assessment of the CPB analysis:**

This is a support task and is being addressed as needed. Have the MC people pursued this?

**Costs:**

**Benefits:**

**Risks:**

**Recommendation:**

Treat as a support task.

---

**4. Development between JIM and d0runjob experts to resolve how additional production paths can be handled in the future.**

Status:

Attached Documentation: none

d0runjob and SAMGrid experts have worked to better separate the responsibilities. The need for new JIM macros for each new processing chain has been removed from SAMGrid. d0runjob now handles this. This has been tested and deployed to the sam v7 MC farms. Future development is now less dependent on changes to SAMGrid.

**Assessment of the CPB analysis:**

As noted, some work has been done.

**Costs:**

**Benefits:**

**Risks:**

**Comments:**

It has been very difficult to find DØ collaborators to test any changes in the system. The testing path has been to role changes out to production and roll back if there are problems. This is a reality of the manpower constraints in the experiment.

**Recommendation:**

DØ should consider if it is actually in a position to test changes or if it is willing to sustain the model of rolling out to production without previous testing. If the roll to production model is preferred, then any requested changes should be kept to a minimum.

---

**5. Ability to run trigsim during Monte Carlo production.**

Status: Not yet implemented.

Attached Documentation: none

We would like to be able to run trigsim as one of the stages of production. This would allow easier trigger studies using a wider range of final states. Additional development/testing by trigsim experts is needed. This is an example of a case issue 4 (above) might resolve such that SAMGrid development might be unnecessary.

**Assessment of the CPB analysis:**



See 4.

**Costs:**

From the perspective of the computing project, this task is "free" in that the functionality is ideally part of the separation of runjob and SAMGrid

From the perspective of DØ, it is likely quite labor intensive to prepare this application for production in a farm environment with an acceptable crash rate.

**Benefits:**

As noted in the comments and perhaps the experiment is pursuing it.

**Risks:**

The risks of not doing this task is to continue running trigger simulations as has been done for the past 5 years. If that has been a bottleneck, then DØ is advised to do the work on their side to get trigSim into production, which would involve making sure that it could be package in such a way as to be suitable for running on a remote farm.

**Recommendation:**

DØ has to figure out if they really need to do this on the farm and are actually prepared to make it happen.

## Production

---

### 1. Skimming in production.

Status: Initial implementation available.

Attached Documentation: (see: **1. Original Skimming document** and **2. Overview of skimming on cab, and requirements for grid version.** )

Have prototype that Daniel is testing (only skimming, not merging). Next step is to solidify prototype and look into merging code. Coupled to the merging of streams task (but are the details different?) If creating and writing multiple streams is the issue here, then the details probably are very different from the merging step.

#### Assessment of the CPB analysis:

##### Costs:

The costs have not been evaluated in detail, but given the number of components that have to change and fact that merging is challenging, it's likely to take significant effort and coordination. The given fact that the SAMGrid system is not yet robustly working for local production, it would be foolhardy to add any new processes to production at this time. Completing this task is far enough in the future that DØ would do well adopt an intermediate solution between running from private areas and full production.

##### Benefits:

There likely is no operational or support savings in terms of human effort. The benefit of pursuing this task is for risk mitigation which could be achieved in other ways.

##### Risks:

Not pursuing this task:

The risk to the experiment is depending on private code/scripts/rcps and the possible loss of expertise. This risk could be mitigated without making skimming part of production by the expedient of taking the first step that would need to be taken anyway—preparing the executable for production running and in archiving scripts and documentation.

A further risk is that the current scripts depend on a certain interface to the resources. That interface is likely to be maintained as long until after the need so that is a minor risk.

*There are very real risks to pursuing this task!* Skimming has a number of stressful failure modes that will be difficult to automate for recovery. Making this part of the production chain is likely to make the production chain more fragile. It doesn't make too much sense to consider doing skimming as a separate SAMGrid production step with separately submitted jobs as we run in that mode now—why replicate a parochial system that is working unless that system has some obvious shortcomings?

##### Comments:

The merging for this task has to be coupled to the other merging issues because it doesn't make any sense to make a series of specific solutions.

The prototype was further from production quality than anyone thought it was.

A WBS structure will be developed for this task in order to try to understand the actual scope of work.

**Recommendation:**

Productize the existing scripts and procedures and executables. Most of the work that is needed to do for that on the DØ side would be necessary to use the grid system anyway so that if in the future we do have a more generic system, much of the ground work on the DØ side is done.

---

**2. Production of CAF Trees.**

Status: Not implemented

Attached Documentation: (see: **3. Overview of caf production on cab, and requirements for a grid version.** )

CAF tree production comes after skimming and is the final production step before analysis. Here, error recovery is critical.

**Assessment of the CPB analysis:**

**Costs:**

The costs have not been evaluated in detail, but given the number of components that have to change, it's likely to take significant effort and coordination. The given fact that the SAMGrid system is not yet robustly working for local production, it is inappropriate to add any new processes to production at this time. Completing this task is far enough in the future that DØ would do well adopt an intermediate solution between running from private areas and full production.

**Benefits:**

There likely is no operational or support savings in terms of human effort. The benefit of pursuing this task is for risk mitigation which could be achieved in other ways.

**Risks:**

Not pursuing this task:

The risk to the experiment is depending on private code/scripts/rcps and the possible loss of expertise. This risk could be mitigated without making making CAF tree production by the expedient of taking the first step that would need to be taken anyway—preparing the executable for production running and in archiving scripts and documentation.

A further risk is that the current scripts depend on a certain interface to the resources. That interface is likely to be maintained as long until after the need so that is a minor risk.

There are very real risks to pursuing this task. Making this part of the production chain is likely to make the production chain more fragile.

**Comments:**

It doesn't make too much sense to consider doing CAF as a separate SAMGrid production step with separately submitted jobs as we run in that mode now—why replicate a parochial system that is working unless that system has some obvious shortcomings?

**Recommendation:**

Productize the existing scripts. Most of the work that is needed to do for that on the DØ side would be necessary to use the grid system anyway so that if in the future we do have a more generic grid system, the ground work on the DØ is done.

---

### 3. Run raw2sim on minbias files.

Status: Not implemented

Attached Documentation: (see: **4. Requirements Document for the Incorporation of DØRaw2Sim into the d0\_runjob/SamGrid Framework** )

This entails processing the "monitor" stream from every run, and writing the output files back into SAM. In principle, error recovery is not crucial (unless every job fails, etc.); a 10-15% loss is acceptable. If we do this on the local farm(s), I assume that the jobs must be able to be submitted through SamGrid.

#### Assessment of the CPB analysis:

##### Costs:

Like TrigSim, with the changes to SAMGrid and d0runjob, the primary work is on the experiment side.

##### Benefits:

Marginal since someone has to run the jobs anyway.

##### Risks:

A risk is that the current scripts depend on a certain interface to the resources. That interface is likely to be maintained as long until after the need so that is a minor risk.

##### Comments:

Who is going to pursue this on the experiment side? If there are currently working scripts, why not put them in CVS and document them?

##### Recommendation:

Most of the work that is needed to do for that on the DØ side would be necessary to use the grid system anyway so that if in the future we do have a more generic system, the ground work on the DØ is done.

---

### 4. Merging of recocert-files

Status: To be implemented This is wanted by DQ group

Attached Documentation:

Merging logic is the same as for merging raw data files..

This one is special in that it will have to use a different executable to perform the merge (its root files instead of the framework stuff).

#### Assessment of the CPB analysis:

All these special cases of merging have to be collapsed into a generic system is handled as a separate project that we could consider after the production operations of the baseline system are working smoothly

Presumably this is currently being done adequately with some set of scripts.

**Costs:**

**Benefits:**

**Risks:**

**Recommendation:**

Productize the existing scripts

---

## 5. Merge more than one stream from Reco (minbias, dilepton).

Status: To be implemented.

Attached Documentation: Need accurate and final requirements list.

Parag has been working on this but not at high priority. Is this a generic problem or are there differences between merging minbias and merging dilepton?

Potentially, file sizes which are very different can affect the merging strategies.

What we have currently is a zerobias stream that Daniel has been working on merging. I believe that it is basically complete (modulo small details). We would like to add a second stream of dilepton-triggered events that would be merged exactly like the zerobias. The only issue that is potentially different is that the dilepton stream may sometimes be very small compared with the zerobias stream. Since it's not currently running, we don't know how small. Obviously, writing a zero-length file is to be avoided... In either case (zerobias or dilepton) 100% success of merging/writing is not necessary, since these streams are purely for diagnostics. Small (few %?) failure rates are tolerable. In detail, if the skimming/merging works, this should be identical, since the same problems apply.

**Assessment of the CPB analysis:**

All these special cases of merging have to be collapsed into a generic system is handled as a separate project that we could consider after the production operations of the baseline system are working smoothly.

**Costs:**

**Benefits:**

**Risks:**

**Comments:**

**Recommendation:**

Generic requirements for merging can be developed, and after the work is scoped out, people can do the cost benefit analysis.

---

## 6. AADST format

status: none

Attached Documentation: none

Currently the B group produces a data format called AADST from the thumbnails. Due to the different techniques used in the B group, it was not practical to use CAF for their analyses.

Currently an individual produces the AADST from the skimmed thumbnails. To remove the need to do this extra step, producing the AADST during production is requested. The logic for this may

well be the same as for the minbias and dimuon merges, perhaps depending on the size of the files from a single partition.

**Assessment of the CPB analysis:**

Progress has been made in putting aadst data into the data handling system.

**Costs:**

**Benefits:**

**Risks:**

**Recommendation:**

productize the existing scripts and executables.

Productize the existing system. Most of the work that is needed to do for that on the DØ side would be necessary to use the grid system anyway so that if in the future we do have a more generic system, the ground work on the DØ is done.

---

## 7. Conversion of d0repro-tools to use SamV7 request system

Status: Not implemented.

Attached Documentation:

Might require extension of SamV7 request system. This is required for safe operation by multiple operators. (And to reduce manual work only Mike D. can do)

**Assessment of the CPB analysis:**

*This is an ongoing task which has been slow to converge. The slowness of progress is instructive as it demonstrates the difficulty of making changes when multiple people are involved and everyone is over committed*

**Costs:**

**Benefits:**

**Risks:**

**Comments:**

---

## Standalone production of recocert output

Status: To be implemented.

Attached Documentation:

This is needed for recovery of failures, although just what failure rate can be tolerated needs to be carefully evaluated. As long as enough events from each luminosity block are processed one can afford to lose some partitions (except for reco, skimming, and CAF-production, where all partitions must be kept).

**Assessment of the CPB analysis:**

It seems unlikely that a tool as coarse as recocert can be used on a luminosity block by luminosity block basis but if that's true, that puts a pretty hard constraint on the merging of recocert files mentioned above.

**Costs:**

Being evaluated in detail. A WBS for this task was developed.

**Benefits:**

Likely none—the current implementation with no error recovery is likely to be adequate and if necessary, someone can run recovery jobs by hand with scripts.

**Risks:**

**Recommendation:**

Productize the existing system. Most of the work that is needed to do for that on the DØ side would be necessary to use the grid system anyway so that if in the future we do have a more generic system, the ground work on the DØ is done.

---

## 8. Minimal Resource Brokering

Status: Functionality is not uniform, among LCG, OSG, or native SAMGrid sites

Attached Documentation: none

SAMGrid should be able to avoid/suppress/delay submission to sites (native, LCG or OSG) sites can't handle additional input. It is suggested to use the number of pending==no-yet-running batch jobs as an indicator. The limit per site may be advertised individually.

**Assessment of the CPB analysis:**

Under evaluation.

**Costs:**

**Benefits:**

**Risks:**

**Recommendation:**

revisit the need after the more pressing tasks are completed

---

## 9. Documentation on adding scaleable forwarding SAMGrid nodes for reprocessing

Status: Not available.

Attached Documentation: none

Parag has done this and it took him a couple of days, involving many steps. DØ would like to be able to do it again, as needed, without his assistance.

**Assessment of the CPB analysis:**

This is being handled as part of the transition from the development team to an operations team.

**Costs:**

**Benefits:**

**Risks:**

**Comments:**

---

## 10. “Fixing”

status: thinking

Attached Documentation: none

Historically, it has been necessary to apply corrections to reconstructed data and this flexibility should be available into the future. Fixing, if we do fixing, will be done by a stand alone tmb-to-tmb, one-to-one (one tmb output stream, all events saved) executable. There is no fixing executable yet. Logically, fixing can be done before or after skimming. Since we have already started skimming, we would probably fix the skimmed thumbnails. Whatever fix we put into a fixer will also be put into a new version of d0reco, so most likely we would only have to fix a subset of the eventual run 2b dataset.

**Assessment of the CPB analysis:**

We have an existence proof of doing fixing with SAMGrid and logically this case is no different from reco in terms of input and less complicated for output since merging is not required. Presumably we can tune up the system to handle this case without further changes. And of course, executable prep is on the DØ side.

**Costs:**

**Benefits:**

**Risks:**

**Comments:**

This case is effectively handled from a SAMGrid perspective, and exe prep is under the auspices of DØ

---

## 11. Grid Analysis

status: searching for coordinator

Attached Documentation: none

A study of the opportunities and costs associated with “analysis on the grid” for DØ has yet to be done in collaboration with the physics groups. Currently, the plan is to identify an individual who will lead the experiment through this decision and then to lead the group charged with implementing it, should it be deemed critical for the analysis effort.

**Assessment of the CPB analysis:**

DØ has not yet completed the needs assessment.

**Costs:**

**Benefits:**

**Risks:**

**Comments:**



Charge from DØ

April 31, 2007

Charge to SAMGrid-DØ Project Manager

Goals of the Project:

4. To enumerate the projects required in order to get DØ datahandling to a state of operational stability for the balance of Run IIb.
  - a. "Projects" might include modifying tasks currently performed in the SAMGrid framework which might be done elsewhere; developing tasks not currently performed in the SAMGrid framework, but might be; and/or adding tasks which might be performed by DØ tools, existing or not.
  - b. "Operational Stability" implies a "reasonable level" of both development and maintenance required of both the SAMGrid team and DØ experimenters. It also implies an acceptable level of effort by DØ personnel in day-to-day DØ datahandling operations.
  - c. The "balance of Run IIb" is defined as through the data-taking period and whatever post-run analysis period is required. A reasonable assumption for this should be specified and what tools are likely required for the post-run period.
5. To analyze the SAMGrid-DØ product architecture and to identify:
  - a. The current functional state of affairs and
  - b. practical solutions to splitting out DØ-specific tasks which might be created given the cost-benefit-risk analysis in 3.
  - c. Specifically: what DØ-specific components from the SAMGrid framework might be migrated into DØ specific tools and what DØ specific components might be retained in the SAMGrid framework.
6. To analyze the cost-benefit-risk tradeoff for each project identified in 1 and 2.
  - a. A "cost" might be:
    - i. CD developer time to maintain SAMGrid, "as is" (which includes possible unforeseen future development) or to modify it as in 2,
    - ii. DØ user development time to modify or maintain existing or new tools;
    - iii. physics opportunity lost by not completing a task in 1; and/or
    - iv. an inefficiency resulting for DØ personnel effort by not completing a task in 1 or 2.
  - b. A "benefit" would be the inverse of the costs in 3a.
  - c. A "risk" would imply the likelihood of future, unforeseen, but necessary development of SAMGrid associated with incomplete migration
  - d. The costs, benefits, and risks should be described and quantified as possible.
7. Make recommendations as to what specific projects from 1 and 2 should be initiated and provide rationale for these recommendations. This should involve an estimate as to DØ's practical ability to carry out any enhanced share of

raymond brock 7/31/07 1:35 PM

**Comment:** I wasn't sure that I meant the same thing as you about "efficient." "Coding" was taken care of in the previous (now) sentence. I mean things like Herb doing hands-on tasks for skimming...that may or may not be an appropriate cost.

development responsibility and suggestions on ways to mitigate concerns in that regard.

8. Make recommendations regarding oversight of the implementation of these projects:
  - a. Who should have this responsibility and
  - b. how long this implementation period should last.
9. Make recommendations of other aspects of DØ datahandling development and/or operations which should receive special future attention, not covered in the above.

A detailed discussion should occur during the DØ Workshop and final recommendations presented to CPB by July 15.